



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/706,656	11/12/2003	Hong Gui	11166 (NCR.0114US)	8622
26890	7590	09/18/2007	EXAMINER	
JAMES M. STOVER NCR CORPORATION 1700 SOUTH PATTERSON BLVD, WHQ3 DAYTON, OH 45479			TIMBLIN, ROBERT M	
			ART UNIT	PAPER NUMBER
			2167	
			MAIL DATE	DELIVERY MODE
			09/18/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

SEP 18 2007

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/706,656
Filing Date: November 12, 2003
Appellant(s): GUI ET AL.

NCR CORPORATION
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 6/25/2007 appealing from the Office action mailed 12/27/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

2003/0093407	Cochrane et al.	5/15/2003
7,035,843	Bellamkonda	4/25/2006

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

Claims 1-5, 13-16, and 22-24 are rejected under 35 U.S.C. 102(a) as being anticipated by Cochrane et al. (U.S. 2003/0093407) (hereinafter Cochrane).

Regarding claim 1, a database system comprising:

a storage (Figure 1, element 104) to store a view containing results of a cube based operation on at least one base table, the view containing a first result set for a group-by on a second grouping set (paragraphs 0036 and 0038); and

a controller (element 118, paragraph 0024), in response to a change to the at least one base table, to:

update the first result set by computing a change to the first result set based on a change in the at least one base table; and update the second result set by computing a change to the second result set based on the change to the first result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second with

summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated.

Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a second result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (the first result set) in order to compute an update to a second result set, meeting the limitation computing a change to the second result set based on the change to the first result set.).

Regarding claim 2, the database system of claim 1, wherein the first grouping set has a first number of grouping attributes, and the second grouping set has a second number of grouping attributes, the first number being greater than the second number (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second (the equivalent of the second result set) with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated).

Regarding claim 3, the database system of claim 2, wherein the view further contains a third result set for a group-by on a third grouping set having a third number of grouping attributes, the third number less than the second number, the controller to further update the third result set by computing a change to the third result set based on the change to the second result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a third result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (a second result set) in order to compute an update to a third result set, meeting the limitation computing a change to the third result set based on the change to the second result set.)).

Regarding claim 4, the database system of claim 1, wherein the view contains results of a group-by cube operation (paragraph 0038).

Regarding claim 5, the database system of claim 1, wherein the view contains results of a group-by partial cube operation (paragraph 0038; a group by grouping sets is the equivalent of a group-by partial cube operation).

Regarding claim 13, a method for use in a database system, comprising: storing a view containing results of a cube-based operation on at least one base table (Figure 1, element 104; the base table is disclosed in paragraph 0035), the view containing result sets for group-bys on respective grouping sets (paragraph 0038); updating a first result set by computing a change to the first result set based on a change in the at least one base table and updating a second result set by computing a change to the second result set based on the change to the first result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation). A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated. Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a second result set would be calculated by

visiting the already materialized view query to obtain pre-computed rows of data (the first result set) in order to compute an update to a second result set, meeting the limitation computing a change to the second result set based on the change to the first result set.).

Regarding claim 14, the method of claim 13, wherein updating the first result set comprises updating the first result set for the group-by on a first grouping set that has a greater number of columns than a second grouping set corresponding to the second result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second (the equivalent of the second result set) with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated).

Regarding claim 15, the method of claim 13, further comprising updating a third result set by computing a change to the third result set based on the change to the second result set (Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some

portions of the materialized view query. Therefore, an update affecting a third result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (a second result set) in order compute an update to a third result set, meeting the limitation computing a change to the second result set based on the change to the first result set).

Regarding claim 16, the method of claim 15, further comprising updating a fourth result set by computing a change to the fourth result set based on the change to the third result set (Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a fourth result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (a third result set) in order to compute an update to a fourth result set, meeting the limitation computing a change to the fourth result set based on the change to the third result set).

Regarding claim 22, an article comprising at least one storage medium containing instructions that when executed cause a database system to:

store a view containing results of a cube-based operation on at least one base table (paragraph 0035), the view containing result sets for group-bys on respective grouping sets (paragraph 0038);

update a first result set by computing a change to the first result set based on a change in the at least one base table; and

update a second result set by computing a change to the second result set based on the change to the first result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated. Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a second result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (the first result set) in order to compute an update to a second result set, meeting the limitation computing a change to the second result set based on the change to the first result set).

Regarding claim 23, the article of claim 22, wherein updating the first result set comprises updating the first result set for the group-by on a first grouping set that has a greater number of columns than a second grouping set corresponding to the result set

Art Unit: 2167

(paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (the first being the equivalent of a first result set), the first displaying summary data of all the columns to be processed, the second (the equivalent of the second result set) with summary data for all the columns of the first, minus one column that is aggregated, the third with yet another column aggregated and so on until all columns are aggregated).

Regarding claim 24, the article of claim 22, wherein the instructions when executed cause the database system to further update a third result set by computing a change to the third result set based on the change to the second result set (paragraphs 0035 and 0038; Cochrane discloses an update to a self-maintainable materialized view in paragraph 0035, and in paragraph 0038 Cochrane discloses wherein the statement that defines the materialized view contains a ROLLUP operation. A rollup operation contains multiple grouping sets (Paragraph 0035 discloses a method in which changes (the equivalent of updates) can be maintained incrementally by processing the updated rows and visiting the underlying tables [of the materialized view query] to recompute some portions of the materialized view query. Therefore, an update affecting a third result set would be calculated by visiting the already materialized view query to obtain pre-computed rows of data (a second result set) in order to compute an update to a

Art Unit: 2167

third result set, meeting the limitation computing a change to the third result set based on the change to the second result set).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 6-12, 17-21, and 25-28 are rejected under 35 U.S.C. 103(a) as being anticipated by Cochrane et al. (U.S. 2003/0093407) (hereinafter Cochrane) in view of Bellamkonda et al. (U.S. 7,035,843) (hereinafter Bellamkonda).

Claims 6-12, 17-21, and 25-28 are rejected under 35 U.S.C. 103(a) as being anticipated by Cochrane et al. (U.S. 2003/0093407) (hereinafter Cochrane) in view of Bellamkonda et al. (U.S. 7,035,843) (hereinafter Bellamkonda).

Regarding claim 6, the database system of claim 1, Cochrane does not explicitly disclose plural access modules and plural storage modules, the access modules to enable parallel access of data in the plural storage modules.

In the same field of endeavor (database analytical processing systems) Bellamkonda discloses plural access modules and plural storage modules, the access modules to enable parallel access of data in the plural storage modules (column 16, line 65-column 17, line 1; slaves in Bellamkonda are the equivalent of access modules. The locations where the rows that each slave processes are the equivalent of plural storage modules). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in

Regarding claim 7, Cochrane does not explicitly disclose wherein the controller is adapted to distribute rows in the first result set across the access modules based on a hash of columns of the second grouping set and at least another column that is assigned to a predefined value.

In the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the controller is adapted to distribute rows in the first

Art Unit: 2167

result set across the access modules based on a hash of columns of the second grouping set and at least another column that is assigned to a predefined value (column 17, lines 25-26; see also Figure 4, column 12, lines 4-9, column 13, lines 43-45 and column 14, lines 39-41. Each grouping set of the rollup calculated in Bellamkonda by this method is a result set. Bellamkonda discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of the second grouping set) in column 12, lines 26-28. The single field disclosed is an extra column not considered in the hash of the second result set (the equivalent of the second grouping set), but is considered in the hash of the first result set (the equivalent of the first grouping set)). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 8, Cochrane discloses a view (paragraph 0035). Cochrane does not explicitly disclose but in the same field of endeavor (database analytical processing

systems) Bellamkonda discloses wherein the view contains results of a cube operation specified by a cube function on plural columns, the at least another column being one of the plural columns of the cube function that is not in the second grouping set (column 12, line 16-column 12, line 28; Bellamkonda discloses a rollup operation which is a cube operation, and further discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of the second grouping set)). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of hashing on columns, including an extra column not present in a second grouping set with Cochrane's teachings of a database system with multiple views with dependencies on other views or tables to obtain a system for updating views that processes an extra column in the hashing of a first grouping set than in a second grouping set for the purpose of a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 9, Cochrane does not explicitly disclose but in the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the view is distributed across the access modules such that plural portions of the view

Art Unit: 2167

reside in respective storage modules, and wherein the rows in the first result set are distributed across the access modules according to the hash to enable:

Each access module to locally perform a merge and aggregate operation on the rows of the first result set to produce rows of the second result set (column 16 line 65-column 17 line 1); and

Each access module to locally merge the rows of the second result set into a respective portion of the view without having to first redistribute the rows of the second result set (column 16 line 65-column 17 line 1, slaves (the equivalent of processing modules) work independently to produce their own grouping sets, which can be merged without rearranging rows). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of access modules to locally perform merge and aggregate operations, without having to first redistribute the rows of the second result set with Cochrane's teachings of a database system with multiple views with dependencies on other views or tables to obtain a system for updating views without having to first redistribute the rows of the second result set for the purpose of a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 10, the database system of claim 1, Cochrane does not explicitly disclose but in the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the controller is adapted to further:

receive a query specifying a group-by operation (column 17, line 61); and

determine whether an answer for the query specifying the group-by operation can be satisfied from the view (column 17, lines 58-61; a determination to decide if more operations need to be done to generate result sets satisfy a query is the same as determining whether an answer for a query can be satisfied from the view). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of receiving a query and determining whether an answer for the query specifying the group-by operation can be satisfied from the view with Cochrane's teachings of a database system with multiple views with dependencies to obtain a database system performance increase in the return of results to a query.

Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 11, the database system of claim 10, Cochrane does not explicitly disclose but in the same field of endeavor (database analytical processing

systems) Bellamkonda discloses wherein the query specifies a group- by operation on grouping sets S, and the view contains result sets for grouping sets C, the controller to determine whether S is a subset of C to determine whether the answer for the query can be satisfied from the view (column 17, lines 58-61; a determination to decide if more operations need to be done to generate result sets satisfy a query is the same as determining whether a group by query specifies grouping sets that are a subset of generated result sets. The result set necessarily created by the disclosed query is the equivalent of S, and records retrieved from the previous rollup operator are the equivalent of C). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of determining whether an answer for the query specifying the group- by operation can be satisfied from the view with Cochrane's teachings of a database system with multiple views with dependencies to obtain a database system performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 12, the database system of claim 11, wherein the controller is adapted to modify a WHERE clause of the query in response to determining that S is a

Art Unit: 2167

subset of C (element 118, paragraph 0024; the database manager supports the functions of the SQL language including WHERE, and upon determining that S is a subset of C, it was well known at the time the invention was made to modify a query based on predetermined information to obtain results with better performance (paragraph 0007)).

Regarding claim 17, Cochrane does not explicitly disclose wherein updating the first result set and second result set are performed in parallel by the plural access modules. In the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein updating the first result set and second result set are performed in parallel by the plural access modules (column 16, line 65-column 17, line 1); slaves in Bellamkonda are the equivalent of access modules. The locations where rows are stored that each slave works on are the equivalent of plural storage modules.). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, lines a need to efficiently evaluate database queries including

hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 18, Cochrane does not explicitly disclose further comprising distributing rows of the first and second result sets across the plural access modules. In the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the controller is adapted to distribute rows in the first result set across the access modules of the second grouping set (column 16, line 65-column 17, line 1). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 19, Cochrane does not explicitly disclose but in the same field of endeavor Bellamkonda discloses wherein the first result set corresponds to a group-by on a first grouping set having N columns, and the second result set corresponds to a

Art Unit: 2167

group-by on a second grouping set having $N-1$ columns, and wherein distributing the first result set to compute the second result set comprises distributing the first result set to compute the second result set comprises distributing the first result set based on a hash of the N columns (column 17, lines 25-26), with the column in the first grouping set not present in the second grouping set assigned a predefined value (column 12, line 16-column 12, line 28; Bellamkonda discloses a rollup operation and further discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of the second grouping set). The result table at each stage is the equivalent of a result set. In rollup operations each stage has one less column than the previous stage, with the single column at each stage being set to a predefined value to indicate aggregation. This meets the limitation of the first result set having N columns and the second result set having $N-1$ columns). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of group-bys with n and $n-1$ columns, processing of hashing on columns, including an extra column not present in a second grouping set with Cochrane's teachings of a database system with multiple views with dependencies on other views or tables to obtain a system for updating views that processes an extra column in the hashing of a first grouping set than in a second grouping set for the purpose of a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-

Art Unit: 2167

23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 20, the method of claim 19, further comprising: Cochrane discloses updating a third result set by computing a change to the third result set based on the change to the second result set, wherein the third result set corresponds to a group-by on a third grouping set (paragraph 0035; in this paragraph Cochrane discloses updates to a plurality of views, some of which are computed based on changes to results sets they depend on. Since grouping sets can have any number of grouping attributes specified, the views disclosed in paragraph 0035 that can be derived from grouping sets meet the limitations of updating a third result set by computing a change to the third result set, wherein the third result set corresponds to a group-by on a third grouping set.)

Cochrane does not explicitly disclose but in the same field of endeavor Bellamkonda discloses wherein the third grouping set has N-2 columns and wherein distributing the second result set across the access modules to compute the third result set is based on a hash of the N columns (column 17, lines 25-26), with the columns in the first grouping set not appearing in the third grouping set each assigned to the predefined value (column 12, line 16-column 12, line 28; Bellamkonda discloses a rollup operation and further discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of

the second grouping set). The result table at each stage is the equivalent of a result set. In rollup operations each stage has one less column than the previous stage, with the single column at each stage being set to a predefined value to indicate aggregation. Therefore, the third stage, the equivalent of the third result set would have the equivalent of $N-2$ columns).

Regarding claim 21, the method of claim 20, wherein storing the view comprises storing a view for a cube operation based on a cube function of the N columns (paragraph 0038).

Regarding claim 25, the database system of claim 1, Cochrane does not explicitly disclose wherein processing is done via plural access modules and plural storage modules, the access modules to enable parallel access of data in the plural storage modules.

In the same field of endeavor (database analytical processing systems) Bellamkonda discloses plural access modules and plural storage modules, the access modules to enable parallel access of data in the plural storage modules (column 16, line 65-column 17, line 1); slaves in Bellamkonda are the equivalent of access modules. The locations where rows are stored that each slave works on are the equivalent of plural storage modules.).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of

Art Unit: 2167

processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 26, Cochrane does not explicitly disclose further comprising distributing rows of the first and second result sets across the plural access modules. In the same field of endeavor (database analytical processing systems) Bellamkonda discloses wherein the controller is adapted to distribute rows in the first result set across the access modules of the second grouping set (column 16, line 65-column 17, line 1). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of processing of plural access modules and plural storage modules with Cochrane's teachings of a database system with multiple views with dependencies to obtain a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 27, Cochrane does not explicitly disclose but in the same field of endeavor Bellamkonda discloses wherein the first result set corresponds to a group-by on a first grouping set having N columns, and the second result set corresponds to a group-by on a second grouping set having N-1 columns, and wherein distributing the first result set to compute the second result set comprises distributing the first result set to compute the second result set comprises distributing the first result set based on a hash of the N columns (column 17, lines 25-26), with the column in the first grouping set not present in the second grouping set assigned a predefined value (column 12, line 16-column 12, line 28; Bellamkonda discloses a rollup operation and further discloses a single field (the equivalent of a column) that is aggregated in the grouping set of the calculations of the next stage (the equivalent of the second grouping set). The result table at each stage is the equivalent of a result set. In rollup operations each stage has one less column than the previous stage, with the single column at each stage being set to a predefined value to indicate aggregation. This meets the limitation of the first result set having N columns and the second result set having N-1 columns). Bellamkonda suggests that the addition of this feature improves the efficiency of query evaluation (column 1, lines 21-23).

Accordingly, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have incorporated Bellamkonda's teachings of group-bys with n and n-1 columns, processing of hashing on columns, including an extra column not present in a second grouping set with Cochrane's teachings of a

database system with multiple views with dependencies on other views or tables to obtain a system for updating views that processes an extra column in the hashing of a first grouping set than in a second grouping set for the purpose of a performance increase in the return of results to a query. Bellamkonda suggests in column 1, lines 21-23, a need to efficiently evaluate database queries including hierarchical cubes. Cochrane suggests a need for a more efficient mechanism for computing database queries in paragraph 0008.

Regarding claim 28, the article of claim 27, wherein storing the view comprises storing a view for a cube operation based on a cube function of the N columns (paragraph 0038).

(10) Response to Argument

Appellant's arguments in the appeal brief filed 6/25/2007 have been fully considered but they are not persuasive.

The Appellant argues the key issue (p. 7-8) that the Cochrane reference does not teach the claimed updating of the second result set (for a group-by on a second grouping set) by computing a change to the second result set based on at he change to the first result set (for a group-by on a first grouping set), where both the first result set and second result set are contained in the same view that contains results of a cube-

based operation on at least one base table. The Examiner disagrees given the following:

Cochrane teaches in their disclosure maintaining materialized views with complex grouping expressions derived from base tables (Cochrane, paragraph 0009). Cochrane further teaches in this paragraph of generating a delta stream to propagate modifications to a materialized view. As an example of creating a materialized view, Cochrane introduces using a CUBE() operation included in a summary table, which is a view containing multiple grouping sets, created on a base table (Cochrane, paragraph 0038 and 0040). As an example of creating a summary table, Cochrane teaches (paragraph 0087 and the following expression) defining a summary table named `ast_hcube` is created by including multiple grouping expressions as well containing a CUBE() operation to define a hierarchical data cube. Furthermore, Cochrane teaches that a CUBE() operation will gather as many result sets as desired. For example, the expression (paragraph 0060) uses `GROUP BY CUBE(loc.city, ti.pgid, month(pdate))` representing three dimensions (i.e. result sets).

The Examiner submits that as a data cube is hierarchical (paragraph 0064), that changes in one dimension (a result set) of a data cube will affect another dimension (a second result set) in the same data cube to ensure data integrity. Cochrane discloses (paragraph 0035, last 4 lines) that materialized views can be maintained *incrementally* by processing the updated rows and visiting the underlying tables to recompute some portions of the materialized query. Cochrane further states (paragraph 0090, last 6 lines) that a change (insert/update/delete) of a single row of the base data [table] can

affect multiple rows in the summary table. The notion of applying incremental changes to a summary table (i.e. containing a data cube) suggests that a change in one result set of the data cube will reflect changes onto a second set of the data cube.

To further note that a change in a second result set is computed based on a change to a first result set, Cochrane suggests (paragraph 0102) the use of an integrity constraint DELETE CASCADE to enforce the propagation of changes from one result set to the next. It is well known in the art that cascading changes (i.e. insert/update/delete) in a database refers to modifying child records based on changes to their related parent records (i.e. within a hierarchy). As a data cube in a summary table in Cochrane is hierarchical in structure, these changes will be applied correspondingly. Cochrane uses the DELETE CASCADE constraint in step I of building a raw delta ("delta" as known in the art, is synonymous with change) from a base table (paragraph 0093). In a second step (Cochrane, paragraph 0103, STEP II) the raw deltas from STEP I are collected and then applied to the summary table (Cochrane, paragraph 0104). The Examiner submits that the deltas in the delta stream carry the effect (i.e. the relations of changes from one set of results based on another set of results) of the DELETE CASCADE constraint and accordingly those changes are applied in the same manner to the summary table. In summary, as changes are cascaded on the base tables of Cochrane, accordingly, the changes are cascaded on the summary table as well.

The Appellant argues on page 9-10, that Cochrane does not provide the multiple levels of updates recited in claim 3 (i.e. computing a change to the third result set based on the change to the second result set).

The Examiner disagrees because as discussed above a data cube in a summary table of Cochrane is hierarchical in structure. The data cube may have multiple dimensions (i.e. result sets) as Cochrane discloses at least three dimensions (i.e. CUBE(A₁₋₃) at the end of the expression in paragraph 0106). As discussed above, changes in a second result are computed based on changes to a first result set. For Cochrane to add a third result set to a data cube and compute changes to it based on changes on a second result set is suggested and within scope of their invention.

The Appellant argues on page 11-12 that none of the keys that are used for partitioning are assigned a predefined value. The Examiner disagrees because in col. 17, line 25-26, Bellamkonda teaches partitioning rows on the grouping identifier. As discussed in the abstract of Bellamkonda, grouping identifiers are used as sorting keys. Furthermore, a single field (also known in the art as a column) serves as a grouping distinguisher (i.e. identifier; Bellamkonda, col. 12 line 25-30), which is given to a grouping set at each particular stage of processing (i.e. a first stage scans the base table and a second stage processes the rollup operation, col. 17 line 10-15). The Examiner submits that Bellamkonda predefines a grouping identifier for each grouping set and then in certain embodiments, rows (i.e. keys) are partitioned on the grouping

Art Unit: 2167

identifier (col. 17 line 26). In summary, as the grouping identifier is given before the partitioning step, it is therefore predefined and subsequently used in partitioning.

The Appellant argues on page 12 of the appeal brief that Bellamkonda does not disclose assigning a predefined value to a column of a grouping set. The Examiner disagrees for the reasons set forth in the rejection above as well as the preceding argument.

The Examiner submits that Bellamkonda teaches assigning a predefined column to a grouping set because Bellamkonda teaches assigning grouping identifiers at each stage of computation. That is, when the grouping set is formed (i.e. base table is scanned), it is given a grouping identifier thereafter for each subsequent processing step. Bellamkonda further specifies assigning a grouping identifier (col. 11 line 48-65) wherein a grouping distinguisher is given at each stage of computation. Therefore, the grouping identifier would be predefined before proceeding to the next stage (i.e. a grouping identifier defined at one stage would be predefined for a second stage).

Art Unit: 2167

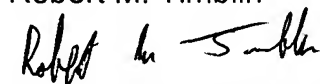
(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Robert M. Timblin



Patent Examiner AU 2167

Conferees:

John R. Cottingham

Mohammad Ali



SPE 2169



JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100